Step 1: Classify Image Type (Structure-focused)

**Goal:** Identify the *container context* (water only, cup, bucket, etc.).

**Approach:** Use GED with **high edge-deletion cost**.

**Idea:** Structure/layout matters more than colour here, since the shape and distribution of edges capture spatial context. Different container types will often have visibly distinct shapes or spatial layouts, which GED can capture through node/edge presence.

Step 2: Determine Water Quality (Colour-focused)

**Goal:** Decide if the water is clean or dirty.

**Approach:** Only compare with graphs of the same type (from step 1), but use GED where **vertex substitution cost (colour change)** is highest.

**Rationale:** Colour difference is a key signal in water clarity/purity, especially when structure is now a controlled variable. Aligns well with how visual indicators (e.g., murkiness, discoloration) are used in human perception.

**Finding node mappings**

During analysis we find that it imperative to find the best possible one-to-one mapping between nodes of two graphs that minimizes the total matching cost. While researching we find that this is called the “Assignment Problem”.  It's a fundamental step in GED computation because:

* Nodes must correspond, so it determines which nodes in Graph A correspond to which nodes in Graph B
* It finds the mapping that results in the lowest total edit operations cost, thereby reducing the costs
* The node mapping also informs how edges should be compared

This is necessary because:

* Water patterns may appear shifted or distorted between images, and this handles special variances
* Some water features may appear/disappear between samples, and these handles feature variations
* Ensures we're comparing the most similar parts of patterns, The most optimal matching

We found that an algorithm already exists to perform this task, he Hungarian algorithm (Munkres algorithm), which:

* Takes a cost matrix where:

1. Rows represent nodes from Graph A
2. Columns represent nodes from Graph B
3. Values represent the cost of matching those nodes

* Finds the assignment that satisfies:

1. Each node is matched to at most one node in the other graph
2. The total cost of all assignments is minimized

All this said and done we have decided to forgo the use of node mappings, because the algorithms present would either be too complex or inadequate, and performing these calculations for all reference data set would be impractical, it will just have to be a limitation of our system.